

Grundlegende Eingabe und Ausgabe in C

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 1

Grundlegende Eingabe und Ausgabe in C

- In diesem Kapitel wollen wir zeigen,
 - ➔ wie man in C Variable einlesen und ausgeben kann
 - ➔ wie die Funktionen **printf (...)** und **scanf (...)** aufgerufen werden
 - ➔ welche Formatanweisungen **printf (...)** und **scanf (...)** verwenden
 - ➔ welche Effekte der Systemeingabepuffer hervorrufen kann.

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 2

Standardausgabefunktion `printf(...)` (1)

- Eine Standardfunktion in C zur **Ausgabe** ist `printf (Stringkonstante, <Variable 1>, <Variable 2>, .... )`
- Dabei ist:
 - * **Stringkonstante** ein konstanter String mit
 - ➔ **Formatierungsangaben** für die Ausgangsvariablen
 - * **<Parameter 1>, <Parameter 2>,** eine **Liste der**
 - ➔ **Ausgangsvariablen**, deren Ausgabeformat in der Stringkonstante angegeben ist.

Beispiel:

`printf ("Betrag in Euro: %f \n", Euro) ;`

- Innerhalb der Stringkonstanten wird die Formatierung der Variablen am vorangestellten **%-Zeichen** erkannt.

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



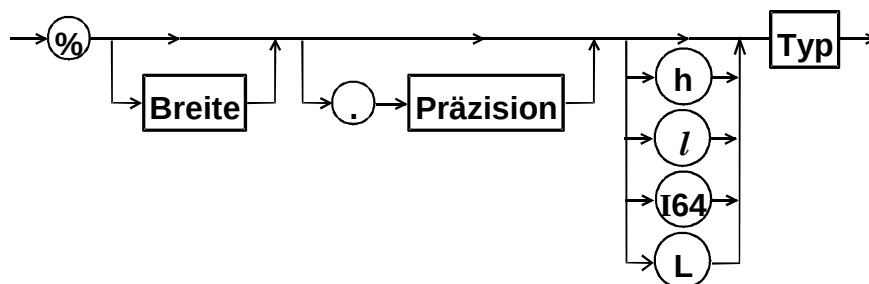
Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 3

19.11.2004

Standardausgabefunktion `printf (...)` (2)

- Im Unterschied zu anderen Programmiersprachen steht in C die Formatanweisung der Variablen im String.

Formatanweisung der Funktion `printf (..)` :



Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard

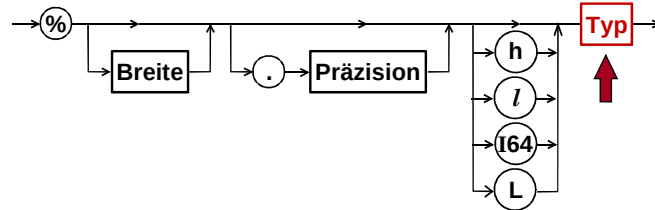


Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 4

19.11.2004

Standardausgabefunktion **printf (...)** (3)

- **Typ** legt den auszugebenden Datentyp fest.



Typ	Datentyp: Ausgabeformat
d	int : Ausgabe dezimal ggf. mit Vorzeichen (-32768 ... 32767)
i	int : wie d
u	int : Ausgabe dezimal ohne Vorzeichen (0 ... 65535)
x	int : Ausgabe hexadezimal ohne Vorzeichen
p	Zeiger (pointer): Adresse des Zeigers (nur für Zeiger zulässig!)
f	Gleitpunktzahl (float): Ausgabe dezimal <u>ohne</u> Exponent: [-]mmmm.dddddd
e	Gleitpunktzahl (float): Ausgabe dezimal <u>mit</u> Exponent: [-]m.dddddd e±xxx genau 1 Ziffer vor dem Dezimalpunkt
c	char : einzelnes Zeichen
s	string (Zeichenkette): Ausgabe bis zum ASCII-Zeichen NUL = '\0'

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd

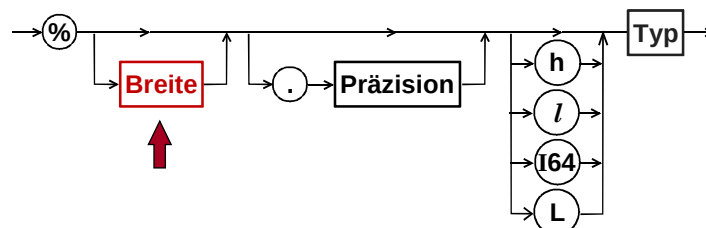


Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 5

19.11.2004

Standardausgabefunktion **printf (...)** (4)

- **Breite** [optional] legt die minimale Feldbreite fest.



- Wenn die Ausgabe **weniger Zeichen** erzeugt als die minimale Feldbreite, werden **zusätzliche Leerzeichen** ausgegeben.
- Wenn die Ausgabe **mehr Zeichen** erzeugt als die minimale Feldbreite, werden die **zusätzlichen Zeichen** ausgegeben.

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd

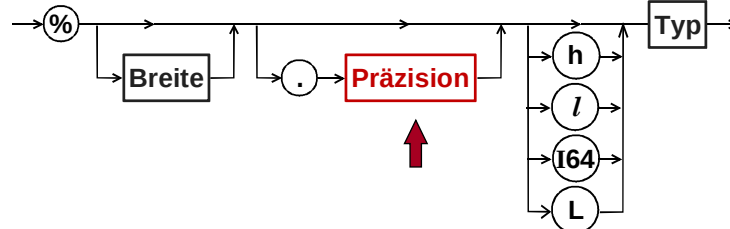


Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 6

19.11.2004

Standardausgabefunktion **printf (...)** (5)

- **Präzision** [*optional*] legt die Genauigkeit fest.



- **String** (Zeichenkette): maximale Anzahl der auszugebenden Zeichen
- **Integer** (ganze Zahl): minimale Anzahl der Ziffern, bei Bedarf werden führende Nullen ergänzt
- **Float** (Typ f oder e): Anzahl der Ziffern nach dem Dezimalpunkt

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard

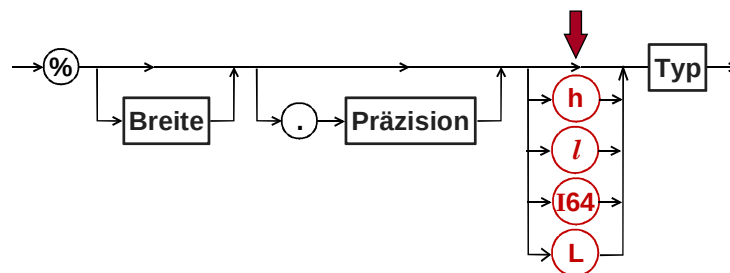


Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 7

19.11.2004

Standardausgabefunktion **printf (...)** (6)

- **Längenangabe** [*optional*] legt den Ausgabedatentyp fest.



Typ	Länge: h, l, I64, L	Ausgabetyt
d, i, u	h	short
	l	long
	I64	hyper
f	l	double
	L	long double

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 8

19.11.2004

Standardeingabefunktion `scanf (...)` (1)

- Eine Standardfunktion in C zur **Eingabe** ist `scanf (Stringkonstante, <Variable 1>, <Variable 2>, ..)`
- Dabei ist:
 - * **Stringkonstante** ein konstanter String mit **Formatierungsangaben** für die Eingangsvariablen
 - * **<Parameter 1>, <Parameter 2>, ...** eine **Liste der Eingangsvariablen mit vorangestelltem Adressoperator &**, deren Eingabeformat in der Stringkonstanten angegeben ist.

Beispiel:

 `scanf ("%f", &DM) ;`


- Innerhalb der Stringkonstanten wird die Formatierung der Variablen am vorangestellten **%-Zeichen** erkannt.

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 9

19.11.2004

Standardeingabefunktion `scanf (...)` (2)

- `scanf` liefert die **Anzahl** der **erkannten** und **zugewiesenen Eingangsvariablen** als **Integerzahl** zurück

`int scanf(Stringkonstante, <Variable 1>, <...>, <...>);`



- Man kann die von `scanf` zurückgegebene Integerzahl verwenden, muss es aber nicht:

Beispiele:

`Richtig = scanf("%f", &DM); /* Ergebnis von scanf wird ausgewertet */`
`scanf ("%d %d", &int1, &dec2); /* Ergebnis wird nicht ausgewertet */`

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 10

19.11.2004

Standardeingabefunktion **scanf (...)** (3)

- Für die **scanf (..)**-Funktion gilt außerdem:
 - * **weiße Leerzeichen** im Formatstring (z.B. **\n**) werden ignoriert.
 - * **normale ASCII-Zeichen** im Formatstring (außer %) werden überlesen
 - * steht im Formatstring ein **Komma zwischen zwei Eingangsvariablen**, so wird bei der Eingabe an dieser Stelle ein Komma als Trennzeichen für die Variablen erwartet und ignoriert

19.11.2004

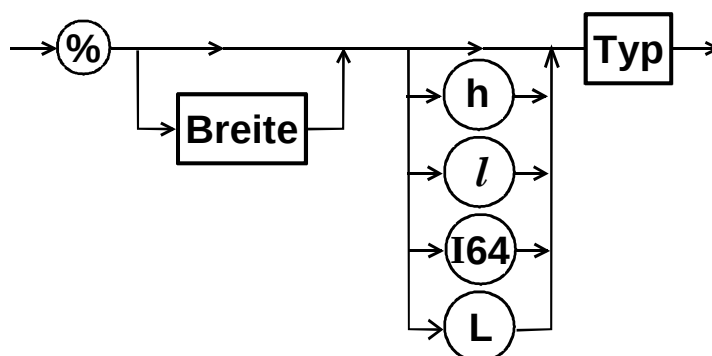
Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 11

Standardeingabefunktion **scanf (...)** (4)

Formatanweisung der Funktion **scanf (..):**



19.11.2004

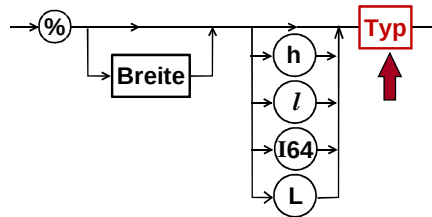
Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 12

Standardeingabefunktion **scanf (...)** (5)

- **Typ** legt den einzugebenden Datentyp fest:



Typ	Datentyp: Eingabeformat
d	Zeiger auf eine Festkommazahl: dezimal Integer (ganze Zahl)
i	Zeiger auf eine Festkommazahl: ganze Zahl • dezimal, wenn die erste Ziffer $\neq 0$ ist • oktal, wenn die erste Ziffer = 0 ist • hexadezimal, wenn die ersten 2 Ziffern = 0x oder 0X sind
f	Zeiger auf eine Gleitkommazahl, zulässig sind • ein optionales Vorzeichen • eine Ziffernfolge ggf. mit Dezimalpunkt • ein optionaler Exponent (e oder E), ein optionales Vorzeichen und eine ganze Zahl Format: [$\pm$] <Vorkomma> [. <Nachkomma>][E e [$\pm$ <Exponent>]]
c	Zeiger auf char : einzelnes Zeichen
s	Zeiger auf string (Zeichenkette): Einlesen bis zum 1. <i>ASCII-Whitespace</i> -Zeichen (<i>Blank</i> = ' ', <i>NUL</i> = '\0', <i>VT</i> = '\v', <i>HT</i> = '\h', <i>line feed</i> = '\n', <i>form feed</i> = '\f')

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd

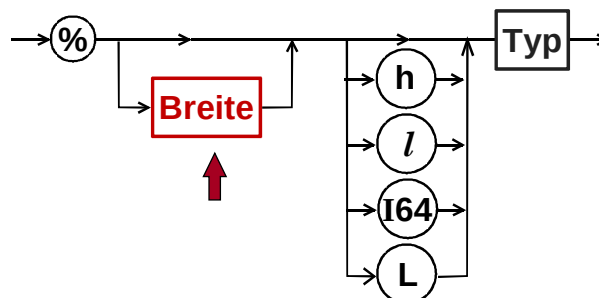


Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 13

19.11.2004

Standardeingabefunktion **scanf (...)** (6)

- **Breite** [optional] legt den maximale Breite des Eingabefeldes fest



- Ein Eingabefeld wird vorzeitig beendet, wenn bei der Eingabe ein unzulässiges Zeichen auftritt.

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 14

19.11.2004

Standardeingabefunktion **scanf (...)** (7)

- **Längenangabe** [*optional*] legt den Eingabedatentyp fest.
- Diese Angabe wirkt nur auf numerische Typen. Sie hat z.B. bei den Typen **c** und **s** keine Wirkung

Typ	Länge: h, l, I64, L	Zeiger auf
d, i, u	h	short
	l	long
	I64	hyper
f	l	double
	L	long double

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagere

Programmieren in ANSI C:
 K3_C-Ein-&Ausgabe.ppt, Seite 15

Tastatureingabe

- Warum bekommt der Benutzer gelegentlich beim Aufruf von **scanf (...)**; keine neue Eingabeaufforderung ?

- Nach dem 1. **Return** beendet **scanf (...)** die Eingabe und löscht die gelesenen Zeichen einschließlich des ↵.
- Das unverbrauchte 2. **Return** (↵) **bleibt im Systempuffer (stdin) stehen** und wird **bei der nächsten Eingabe** als **Endezeichen** interpretiert.

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagere

Programmieren in ANSI C:
 K3_C-Ein-&Ausgabe.ppt, Seite 16

Löschen des Eingabepuffers mit `fflush(stdin)`

- Warum bekommt der Benutzer gelegentlich beim Aufruf von `scanf (...)` ; keine neue Eingabeaufforderung ?
- Wenn bei der letzten Eingabe noch Zeichen im **Betriebssystempuffer** `stdin` übrig waren, so bleiben diese im Puffer stehen und werden beim **nächsten Aufruf** von `scanf` gelesen.
- `scanf (...)` interpretiert ein überflüssiges ↵ (**Return**) als **Eingabeende** und liest ein „falsches“ Zeichen.
- Will man dies verhindern, so muss man den Eingabepuffer vor dem erneuten Aufruf von `scanf(..)` ; mit ➡ `fflush(stdin)` ; löschen

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 17

Löschen des des *MS DOS*-Bildschirms unter *MSVC 6.0*

- Zum Löschen des Bildschirms muss man auf folgende *MS DOS*-Betriebssystembefehle der Bibliotheken `stdio.h` bzw. `stdlib.h` zurückgreifen:
- `fflush(stdin)` ; /* Muss vor "system"-Kommandos ausgeführt werden! */
- ➡ `system("cls")` ; /* Bildschirm löschen */
- Das Löschen des Bildschirms ist in *MSVC* im Unterschied zu früheren *MS DOS IDEs* nur sehr selten notwendig, weil in *MSVC* **jedes Programm in einem neuen Fenster** gestartet wird

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K3_C-Ein-&Ausgabe.ppt, Seite 18